

**IMPROVED COMPUTING ARCHITECTURE AND RELATED SYSTEM AND METHOD**

Publication number: JP2006518057 (T)

Publication date: 2006-08-03

Inventor(s):

Applicant(s):

Classification:

- international: G06F15/80; G06F9/30; G06F9/38; G06F9/445; G06F9/46; G06F15/78; G06F15/76; G06F9/30; G06F9/38; G06F9/445; G06F9/46


- European: G06F9/3884


Application number: JP20050502224T 20031031


Priority number(s): US20030422503P 20021031; US20030683929 20031009; US20030683932 20031009; US20030684053 20031009; US20030684057 20031009; US20030684102 20031009; WO20030634557 20031031


Also published as:

 WO2004042560 (A2)

 WO2004042560 (A3)

 WO2004042574 (A2)

 WO2004042574 (A3)

 WO2004042569 (A2)

more &gt;&gt;

Abstract not available for JP 2006518057 (T)

Abstract of corresponding document: **WO 2004042560 (A2)**

A peer-vector machine includes a host processor and a hardwired pipeline accelerator. The host processor executes a program, and, in response to the program, generates host data, and the pipeline accelerator generates pipeline data from the host data. Alternatively, the pipeline accelerator generates the pipeline data, and the host processor generates the host data from the pipeline data. Because the peer-vector machine includes both a processor and a pipeline accelerator, it can often process data more efficiently than a machine that includes only processors or only accelerators. For example, one can design the peer-vector machine so that the host processor performs decision-making and non-mathematically intensive operations and the accelerator performs non-decision-making and mathematically intensive operations. By shifting the mathematically intensive operations to the accelerator, the peer-vector machine often can, for a given clock frequency, process data at a speed that surpasses the speed at which a processor-only machine can process the data.

.....  
Data supplied from the **esp@cenet** database --- Worldwide

(19) 日本国特許庁 (JP)

(12) 公表特許公報 (A)

(11) 特許出願公表番号

特表2006-518057

(P2006-518057A)

(43) 公表日 平成18年8月3日 (2006.8.3)

(51) Int. Cl.

G06F 15/80 (2006.01)

F I

G06F 15/80

テーマコード (参考)

審査請求 未請求 予備審査請求 未請求 (全 25 頁)

(21) 出願番号 特願2005-502224 (P2005-502224)  
 (86) (22) 出願日 平成15年10月31日 (2003.10.31)  
 (85) 翻訳文提出日 平成17年6月16日 (2005.6.16)  
 (86) 国際出願番号 PCT/US2003/034557  
 (87) 国際公開番号 W02004/042560  
 (87) 国際公開日 平成16年5月21日 (2004.5.21)  
 (31) 優先権主張番号 60/422,503  
 (32) 優先日 平成14年10月31日 (2002.10.31)  
 (33) 優先権主張国 米国 (US)  
 (31) 優先権主張番号 10/683,929  
 (32) 優先日 平成15年10月9日 (2003.10.9)  
 (33) 優先権主張国 米国 (US)  
 (31) 優先権主張番号 10/683,932  
 (32) 優先日 平成15年10月9日 (2003.10.9)  
 (33) 優先権主張国 米国 (US)

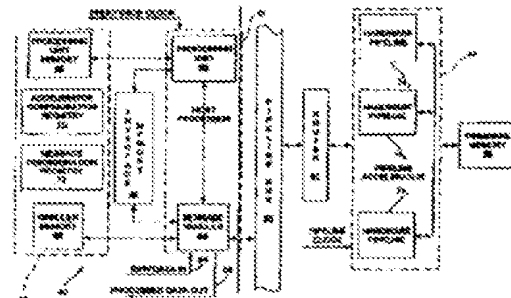
(71) 出願人 504242618  
 ロッキード マーティン コーポレーション  
 アメリカ合衆国 メリーランド州 208  
 17-1803 ベセスダ ロックレッジ  
 ドライブ 6801  
 (74) 代理人 100083932  
 弁理士 廣江 武典  
 (74) 代理人 100129698  
 弁理士 武川 隆宣  
 (74) 代理人 100129676  
 弁理士 ▲高▼荒 新一  
 (74) 代理人 100130074  
 弁理士 中村 繁元

最終頁に続く

(54) 【発明の名称】 改善された計算アーキテクチャ、関連システム、並びに、方法

## (57) 【要約】

ピアーベクトル・マシンはホストプロセッサとハードウェアに組み込まれたパイプライン加速器を含む。ホストプロセッサはプログラムを実行し、該プログラムに応じてホスト・データを生成し、そしてパイプライン加速器はそのホスト・データからパイプライン・データを生成する。代替的には、パイプライン加速器はパイプライン・データを生成し、ホストプロセッサはそのパイプライン・データからホスト・データを生成する。ピアーベクトル・マシンはプロセッサ及びパイプライン加速器の両方を含むので、しばしば、プロセッサだけ或は加速器だけを含むマシンよりもより効率的にデータを処理できる。例えば、ホストプロセッサが意思決定を実行すると共に数学的に非集中的な演算を実行し、そして加速器が非意思決定を実行すると共に数学的に集中的な演算を実行するようにピアーベクトル・マシンを設計できる。数学的に集中的な演算を加速器にシフトすることによって、ピアーベクトル・マシンは、しばしば、所与のクロック周波数で、プロセッサだけのマシンがデータを処理できる速度を凌ぐ速度でデータを処理できる。



## 【特許請求の範囲】

## 【請求項1】

ビアーベクトル・マシンであって、  
プログラムを実行するように動作でき、そのプログラムに応じて第1ホスト・データを生成するように動作できるホストプロセッサと、

前記ホストプロセッサと結合されて、前記第1ホスト・データを受信して、該第1ホスト・データから第1パイプライン・データを生成するように動作できるパイプライン加速器と、

を備えるビアーベクトル・マシン。

## 【請求項2】

10

前記ホストプロセッサが、  
第2データを受信し、  
前記第2データから前記第1ホスト・データを生成するように更に動作できる、請求項1に記載のビアーベクトル・マシン。

## 【請求項3】

前記ホストプロセッサが、  
前記パイプライン加速器から前記第1パイプライン・データを受信し、  
前記第1パイプライン・データを処理するように更に動作できる、請求項1に記載のビアーベクトル・マシン。

## 【請求項4】

20

前記ホストプロセッサが、  
前記パイプライン加速器から前記第1パイプライン・データを受信し、  
前記第1パイプライン・データから前記第1ホスト・データを生成するように更に動作できる、請求項1に記載のビアーベクトル・マシン。

## 【請求項5】

前記ホストプロセッサと結合されると共に前記パイプライン加速器と結合され、第1メモリ区分を有するインターフェース・メモリを更に含み、

前記ホストプロセッサが、  
前記第1メモリ区分に前記第1ホスト・データを記憶し、  
前記第1ホスト・データを前記第1メモリ区分から前記パイプライン加速器に提供する  
ように動作できる、請求項1に記載のビアーベクトル・マシン。

30

## 【請求項6】

前記ホストプロセッサと結合されると共に前記パイプライン加速器と結合され、第1及び第2のメモリ区分を有するインターフェース・メモリを更に含み、

前記ホストプロセッサが、  
前記第1メモリ区分に前記第1ホスト・データを記憶し、  
前記第1メモリ区分から前記パイプライン加速器に前記第1ホスト・データを提供し

、  
前記パイプライン加速器から前記第1パイプライン・データを受信し、  
前記第2メモリ区分に前記第1パイプライン・データを記憶し、  
前記第2メモリ区分から前記ホストプロセッサまで前記第1パイプライン・データを  
検索し、

40

前記第1パイプライン・データを処理するように動作できる、請求項1に記載のビアーベクトル・マシン。

## 【請求項7】

前記ホストプロセッサが前記パイプライン加速器を構成するように動作できる、請求項1に記載のビアーベクトル・マシン。

## 【請求項8】

前記パイプライン加速器が、プログラマブル論理集積回路を含む、請求項1に記載のビアーベクトル・マシン。

50

## 【請求項 9】

ビアーベクトル・マシンであって、  
第 1 パイプライン・データを生成するように動作できるパイプライン加速器と、  
前記パイプライン加速器と結合されると共に、プログラムを実行するように動作でき、  
そのプログラムに応じて第 1 パイプライン・データを受信し、該第 1 パイプライン・データから第 1 ホスト・データを生成するように動作できるホストプロセッサと、  
を備えるビアーベクトル・マシン。

## 【請求項 10】

前記パイプライン加速器が、  
第 2 データを受信し、  
前記第 2 データから前記第 1 パイプライン・データを生成するように更に動作できる、  
請求項 9 に記載のビアーベクトル・マシン。

10

## 【請求項 11】

前記パイプライン加速器が、  
前記ホストプロセッサから前記第 1 ホスト・データを受信し、  
前記第 1 ホスト・データを処理するように更に動作できる、請求項 9 に記載のビアーベクトル・マシン。

## 【請求項 12】

前記パイプライン加速器が、  
前記ホストプロセッサから前記第 1 ホスト・データを受信し、  
前記第 1 ホスト・データから前記第 1 パイプライン・データを生成するように更に動作できる、請求項 9 に記載のビアーベクトル・マシン。

20

## 【請求項 13】

前記パイプライン加速器と結合されると共に前記ホストプロセッサと結合され、第 1 メモリ区分を有するインターフェース・メモリを更に含み、  
前記ホストプロセッサが、  
前記第 1 メモリ区分に前記パイプライン加速器からの前記第 1 パイプライン・データを記憶し、  
前記第 1 メモリ区分から前記第 1 パイプライン・データを検索するように動作できる、  
請求項 9 に記載のビアーベクトル・マシン。

30

## 【請求項 14】

前記パイプライン加速器と結合されると共に前記ホストプロセッサと結合され、第 1 及び第 2 のメモリ区分を有するインターフェース・メモリを更に含み、  
前記ホストプロセッサが、  
前記第 1 メモリ区分に前記パイプライン加速器からの前記第 1 パイプライン・データを記憶し、  
前記第 1 メモリ区分から前記第 1 パイプライン・データを検索し、  
前記第 2 メモリ区分に前記第 1 ホスト・データを記憶し、  
前記第 2 メモリ区分から前記パイプライン加速器まで前記第 1 ホスト・データを提供するように動作でき、  
前記パイプライン加速器が前記第 2 メモリ区分から受信した前記第 1 ホスト・データを処理するように動作できる、請求項 9 に記載のビアーベクトル・マシン。

40

## 【請求項 15】

前記ホストプロセッサが前記パイプライン加速器を構成するように動作できる、請求項 9 に記載のビアーベクトル・マシン。

## 【請求項 16】

システムであって、  
生データを生成するように動作できる装置と、  
前記装置と結合されると共に、プログラムを実行するように動作でき、前記プログラムに応じて前記生データからホスト・データを生成するように動作できるホストプロセッサ

50

と、

前記ホストプロセッサと結合されると共に、前記ホスト・データを受信して該ホスト・データからパイプライン・データを生成するように動作できるパイプライン加速器と、を備えるシステム。

【請求項 17】

システムであって、

生データを生成するように動作できる装置と、

前記装置と結合されると共に、前記生データからパイプライン・データを生成するように動作できるパイプライン加速器と、

前記パイプライン加速器と結合されると共に、プログラムを実行するように動作でき、前記プログラムに応じて前記パイプライン・データを受信し、そして前記パイプライン・データからホスト・データを生成するように動作できるホストプロセッサと、を備えるシステム。 10

【請求項 18】

方法であって、

ホストプロセッサによってプログラムを実行することによって第 1 ホスト・データを生成し、

パイプライン加速器によって前記第 1 ホスト・データから第 1 パイプライン・データを生成することを含む方法。

【請求項 19】

生データを受信し、

前記第 1 ホスト・データを生成することが、前記生データから前記第 1 ホスト・データを生成することを更に含む、請求項 18 に記載の方法。 20

【請求項 20】

前記第 1 ホスト・データを生成することが、前記第 1 パイプライン・データから前記第 1 ホスト・データを生成することを含む、請求項 18 に記載の方法。

【請求項 21】

前記ホストプロセッサによって前記プログラムを実行することによって前記第 1 パイプライン・データから第 2 ホスト・データを生成することを更に含む、請求項 18 に記載の方法。 30

【請求項 22】

前記ホストプロセッサによって前記プログラムを実行することによって前記パイプライン加速器を構成することを更に含む、請求項 18 に記載の方法。

【請求項 23】

パイプライン加速器によって第 1 パイプライン・データを生成し、

ホストプロセッサによってプログラムを実行することによって前記第 1 パイプライン・データから第 1 ホスト・データを生成することを含む方法。

【請求項 24】

生データを受信することを更に含む、

前記第 1 パイプライン・データを生成することが、前記生データから前記第 1 パイプライン・データを生成することを含む、請求項 23 に記載の方法。 40

【請求項 25】

前記第 1 パイプライン・データを生成することが、前記第 1 ホスト・データから前記第 1 パイプライン・データを生成することを含む、請求項 23 に記載の方法。

【請求項 26】

前記パイプライン加速器によって前記第 1 ホスト・データから第 2 パイプライン・データを生成することを更に含む、請求項 23 に記載の方法。

【請求項 27】

前記ホストプロセッサによって前記プログラムを実行することによって前記パイプライン加速器を構成することを更に含む、請求項 23 に記載の方法。 50

【発明の詳細な説明】

【技術分野】

【0001】

＜優先権の請求＞

この出願は、下記の特許文献1に対する優先権を請求するものであり、引用することによってここに合体させる。

【特許文献1】米国仮出願第60/422,503号（2002年10月31日出願）

【0002】

＜関連出願の相互参照＞

この出願は、「改善された計算アーキテクチャを有する計算マシン、関連システム、並びに、方法」と題された下記の特許文献2、「改善された計算アーキテクチャ用パイプライン加速器、関連システム、並びに、方法」と題された下記の特許文献3、「プログラマブル回路、関連計算マシン、並びに、方法」と題された下記の特許文献4、「多数パイプライン・ユニットを有するパイプライン加速器、関連計算マシン、並びに、方法」と題された下記の特許文献5と関連し、これら特許文献は全て2003年10月9日出願され、共通の所有者を有し、引用することによってここに合体させる。

【特許文献2】米国出願第10/684,053号

【特許文献3】米国出願第10/683,929号

【特許文献4】米国出願第10/684,057号

【特許文献5】米国出願第10/683,932号

【背景技術】

【0003】

比較的大量のデータを比較的短い期間で処理する通常の計算アーキテクチャは、処理負担を分担する多数の相互接続プロセッサを含む。処理負担を分担することによって、これら多数のプロセッサは、しばしば、所与のクロック周波数で単一プロセッサができるものよりも迅速にデータを処理できる。例えば、これらプロセッサの各々はデータの各部分を処理できるか、或は、処理アルゴリズムの各部分を実行できる。

【0004】

図1は、多数プロセッサ・アーキテクチャを有する従来の計算マシン10の概略ブロック図である。この計算マシン10は、マスター・プロセッサ12と、相互に通信すると共に該マスター・プロセッサとバス16を介して通信する共同プロセッサ14<sub>1</sub>〜14<sub>n</sub>と、遠隔装置（図1では不図示）から生データを受け取る入力ポート18と、該遠隔装置に処理データを提供する出力ポート20とを含む。また、計算マシン10はマスター・プロセッサ12に対するメモリ22と、共同プロセッサ14<sub>1</sub>〜14<sub>n</sub>に対する各メモリ24<sub>1</sub>〜24<sub>n</sub>と、マスター・プロセッサ及び共同プロセッサがバス16を介して共有するメモリ26とを含む。メモリ22はマスター・プロセッサ12に対するプログラム及び作業メモリの双方の役割を果たし、各メモリ24<sub>1</sub>〜24<sub>n</sub>は各共同メモリ14<sub>1</sub>〜14<sub>n</sub>に対するプログラム及び作業メモリの双方の役割を果たす。共有されたメモリ26は、マスター・プロセッサ12及び共同プロセッサ14がそれらの間でデータを転送すること、ポート18を介して遠隔装置からデータを転送すること、ポート20を介して遠隔装置にデータを転送することを可能としている。またマスター・プロセッサ12及び共同プロセッサ14は、マシン10が生データを処理する速度を制御する共通クロック信号を受け取る。

【0005】

一般に、計算マシン10は、マスター・プロセッサ12及び共同プロセッサ14の間で生データの処理を効果的に分割する。ソナー・アレイ（図5）等の遠隔ソース（図1では不図示）は、ポート18を介して、生データに対する先入れ先出し（FIFO）バッファ（不図示）として作用する共有メモリ26の1つの区分に生データをロードする。マスター・プロセッサ12はバス16を介してメモリ26から生データを検索して、マスター・プロセッサ及び共同プロセッサ14はその生データを処理して、バス16を介して必要に応じてデータをそれらの間に転送する。マスター・プロセッサ12はその処理データを共

有メモリ26内に規定された別のFIFOバッファ（不図示）にロードし、遠隔ソースがポート20を介してこのFIFOからその処理データを検索する。

#### 【0006】

演算例において、計算マシン10は生データに対する $n+1$ 個の各演算を順次実行することによって該生データを処理し、これら演算は一体的に高速フーリエ変換（FFT）等の処理アルゴリズムを構成する。より詳細には、マシン10はマスター・プロセッサ12及び共同プロセッサ14からのデータ処理パイプラインを形成する。クロック信号の所与の周波数で、そうしたパイプラインはしばしばマシン10が単一プロセッサのみを有するマシンよりも高速に生データを処理することを可能としている。

#### 【0007】

メモリ26内における生データFIFO（不図示）からの生データ検索後、マスター・プロセッサ12はその生データに対して三角関数等の第1番演算を実行する。この演算は第1番結果を生み出し、それをプロセッサ12がメモリ26内に規定された第1番結果FIFO（不図示）に記憶する。典型的には、プロセッサ12はメモリ22内に記憶されたプログラムを実行し、そのプログラムの制御の下で上述した動作を実行する。プロセッサ12はメモリ22を作業メモリとしても使用し得て、当該プロセッサが第1番演算の中間期間に生成するデータを一時的に記憶する。

#### 【0008】

次に、メモリ26内における第1番結果FIFO（不図示）からの第1番結果検索後、共同プロセッサ14はその第1番結果に対して対数関数等の第2番演算を実行する。この第2番演算は第2番結果を生み出し、それを共同プロセッサ14がメモリ26内に規定された第2番結果FIFO（不図示）に記憶する。典型的には、共同プロセッサ14はメモリ24内に記憶されたプログラムを実行し、そのプログラムの制御の下で上述した動作を実行する。共同プロセッサ14はメモリ24を作業メモリとしても使用し得て、当該共同プロセッサが第2番演算の中間期間に生成するデータを一時的に記憶する。

#### 【0009】

次に共同プロセッサ24<sub>1</sub>、24<sub>2</sub>は、共同プロセッサ24<sub>1</sub>に対して先に議論されたものと同様に、（第2番結果～第 $(n-1)$ 番）結果に対して（第3番演算～第 $n$ 番）演算を順次実行する。

#### 【0010】

共同プロセッサ24<sub>1</sub>によって実行される第 $n$ 番演算は最終結果、即ち処理データを生み出す。共同プロセッサ24<sub>1</sub>はその処理データをメモリ26内に規定された処理データFIFO（不図示）内にロードし、遠隔装置（図1では不図示）がこのFIFOからその処理データを検索する。

#### 【0011】

マスター・プロセッサ12及び共同プロセッサ14は処理アルゴリズムの種々の演算を同時に実行するので、計算マシン10は、しばしば、種々の演算を順次実行する単一プロセッサを有する計算マシンよりも生データを高速に処理することができる。詳細には、単一プロセッサは、生データから成る先行集合に対する全 $(n+1)$ 個の演算を実行するまで、生データから成る新しい集合を検索できない。しかし、以上に議論したパイプライン技術を用いて、マスター・プロセッサ12は第1演算だけを実行後に生データから成る新しい集合を検索できる。結果として、所与のクロック周波数でこのパイプライン技術は、単一プロセッサ・マシン（図1では不図示）と比較して約 $n+1$ 倍だけマシン10が生データを処理する速度を増大することができる。

#### 【0012】

代替的には、計算マシン10は、生データに対するFFT等の処理アルゴリズムの $(n+1)$ 例を同時に実行することによって該生データを並列して処理し得る。即ち、もしそのアルゴリズムが先行する例において先に記載されたような $(n+1)$ 個の順次演算を言え、マスター・プロセッサ12及び共同プロセッサ14の各々は生データからそれぞれが成る各集合に対して、順次、全 $(n+1)$ 個の演算を実行する。その結果として、所与

のクロック周波数で、先のパイプライン技術と同様のこの並列処理技術は、単一プロセッサ・マシン（図1では不図示）と比較して約  $n+1$  倍だけマシン10が生データを処理する速度を増大することができる。

#### 【0013】

残念ながら、計算マシン10は単一プロセッサ・計算マシン（図1では不図示）と比べてより迅速にデータを処理できるが、マシン10のデータ処理速度はしばしばプロセッサ・クロックの周波数より非常に小さい。詳細には、計算マシン10のデータ処理速度はマスター・プロセッサ12及び共同プロセッサ14がデータ処理するのに必要な時間によって制限される。簡略化のため、この速度制限の例はマスター・プロセッサ12と連携して議論されているが、この議論は共同プロセッサ14にも適用されることを理解して頂きたい。先に議論されたように、マスター・プロセッサ12は所望の方式でデータを操作すべくプロセッサを制御するプログラムを実行する。このプログラムはプロセッサ12が実行する複数の命令から成るシーケンスを含む。残念ながら、プロセッサ12は典型的には単一命令を実行するために多数のクロック・サイクルを必要とし、そしてしばしばデータの単一値を処理すべく多数の命令を実行しなければならない。例えば、プロセッサ12が第1データ値A（不図示）を第2データ値B（不図示）で乗算することを仮定する。第1クロック・サイクル中、プロセッサ12はメモリ22から乗算命令を検索する。第2及び第3クロック・サイクル中、プロセッサ12はメモリ26からA及びBをそれぞれ検索する。第4クロック・サイクル中、プロセッサ12はA及びBを乗算し、そして第5クロック・サイクル中に結果としての積をメモリ22或は26に記憶するか、或は、その結果としての積を遠隔装置（不図示）に提供する。これは最良ケースのシナリオであり、その理由は多くの場合にプロセッサ12はカウンタの初期化及び閉鎖等のオーバーヘッド・タスクに対して付加的なクロック・サイクルを必要とするからである。それ故に、よくてもプロセッサ12はA及びBを処理すべく5クロック・サイクルを必要とするか、或は、1データ値当たり平均2.5クロック・サイクルを必要とする。

#### 【0014】

結果として、計算マシン10がデータを処理する速度は、しばしば、マスター・プロセッサ12及び共同プロセッサ14を駆動するクロックの周波数より非常に低い。例えば、もしプロセッサ12は1.0ギガヘルツ（GHz）でクロックされるが、1データ値当たり平均2.5クロック・サイクルを必要とすれば、効果的なデータ処理速度は  $(1.0 \text{ GHz}) / 2.5 = 0.4 \text{ GHz}$  と同等である。この効果的なデータ処理速度は、しばしば、1秒当たり演算数の単位で特徴付けされる。それ故に、この例において、1.0 GHzのクロック速度で、プロセッサ12は0.4ギガ演算数/秒（Gops）で使用限界が定められる。

#### 【0015】

図2は、所与クロック周波数で且つしばしば該パイプラインがクロックされる速度と略同一速度で、プロセッサが可能であるよりは高速で典型的にはデータを処理できるハードウェアに組み込まれたデータ・パイプライン30のブロック図である。パイプライン30は、プログラム命令を実行することなく、各データに対する各演算を各々が実行する演算子回路32<sub>1</sub>〜32<sub>n</sub>を含む。即ち、所望の演算は回路32内に「書き込み」が為されて、それがプログラム命令の必要性なしに自動的にその演算を具現化するように為す。プログラム命令の実行と関連されたオーバーヘッドを減ずることによって、パイプライン30は所与のクロック周波数でプロセッサが可能であるよりは単位秒当たりより多くの演算を典型的には実行する。

#### 【0016】

例えば、パイプライン30は所与のクロック周波数でプロセッサが可能であるよりは高速で以下の数式1をしばしば解くことができる。

$$Y(x_i) = (5x_i + 3)2^{i-1}$$

ここで、 $x_i$ は複数の生データ値から成るシーケンスを表す。この例において、演算子回路32<sub>1</sub>は  $5x_i$  を計算する乗算器であり、回路32<sub>2</sub>は  $5x_i + 3$  を計算する加算器であ



り、そして回路32<sub>n</sub> (n = 3) は  $(5x_k + 3) 2^{k-1}$  を計算する乗算器である。

【0017】

第1クロック・サイクルk = 1中、回路32<sub>1</sub>はデータ値x<sub>1</sub>を受け取って、それを5で乗じて、5x<sub>1</sub>を生成する。

【0018】

第2クロック・サイクルk = 2中、回路32<sub>2</sub>は回路32<sub>1</sub>から5x<sub>1</sub>を受け取って、3を加えて、5x<sub>1</sub> + 3を生成する。またこの第2クロック・サイクル中に回路32<sub>1</sub>は5x<sub>2</sub>を生成する。

【0019】

第3クロック・サイクルk = 3中、回路32<sub>3</sub>は回路32<sub>2</sub>から5x<sub>1</sub> + 3を受け取って、2<sup>k-1</sup>で乗じて（効果としては、x<sub>1</sub>だけ5x<sub>1</sub> + 3を右シフトする）、第1結果(5x<sub>1</sub> + 3) 2<sup>k-1</sup>を生成する。またこの第3クロック・サイクル中に回路32<sub>1</sub>は5x<sub>3</sub>を生成し、回路32<sub>2</sub>は5x<sub>2</sub> + 3を生成する。

【0020】

このようにしてパイプライン30は、全ての生データ値が処理されるまで、引き続く生データ値x<sub>k</sub>の処理を続行する。

【0021】

結果として、生データ値x<sub>1</sub>の受け取り後の2つのクロック・サイクルの遅延、即ち、この遅延はパイプライン30の待ち時間としばしば呼称され、パイプラインは結果(5x<sub>1</sub> + 3) 2<sup>k-1</sup>を生成し、その後、各クロック・サイクル毎に1つの結果を生成する。

【0022】

待ち時間を無視して、パイプライン30はこうしてクロック速度と同等のデータ処理速度を有する。比較して、マスター・プロセッサ12及び共同プロセッサ14（図1）が先の例におけるようにクロック速度の0.4倍であるデータ処理速度を有すると仮定すれば、パイプライン30は、所与のクロック速度で、計算マシン10（図1）よりも2.5倍高速でデータを処理できる。

【0023】

更に図2で参照されるように、設計者はフィールド・プログラマブル・ゲート・アレイ（FPGA）等のプログラマブル・ロジックIC（PLIC）にパイプライン30を具現化することを選ぶ可能性があり、その理由はPLICが特殊用途IC（ASIC）が為すよりも多くの設計及び変更の柔軟性を許容するからである。PLIC内にハードウェアに組み込まれた接続を構成するため、設計者はPLIC内に配置された相互接続構成レジスタを単に所定バイナリー状態に設定する。全てのこうしたバイナリー状態の組み合わせはしばしば「ファームウェア」と呼称される。典型的には、設計者はこのファームウェアをPLICと結合された不揮発性メモリ（図2では不図示）内にロードする。PLICを「ターンオン」すると、それはファームウェアをそのメモリから相互接続構成レジスタにダウンロードする。それ故に、PLICの機能を変更すべく、設計者は単にそのファームウェアを変更して、PLICがその変更されたファームウェアを相互接続構成レジスタにダウンロードすることを可能とする。ファームウェアを単に変更することによってPLICを変更する能力は、モデル作成段階中や「フィールド内」にパイプライン30をアップグレードするために特に有用である。

【0024】

残念ながら、ハードウェアに組み込まれたパイプライン30は、典型的には、全てのアルゴリズムを実行することができるわけではなく、特に、重要な意思決定を引き起こすアルゴリズムは実行できない。プロセッサは、典型的には、意思決定命令（例えば、「もしAであれば、Bへ行き、またCへ行く」のような、条件命令）を、比較する長さの演算命令（例えば、「A + B」）を実行できる程に高速に実行できる。しかしパイプライン30は、比較的単純な決定（例えば、「A > B？」）を為し得るが、典型的には比較的複雑な決定（例えば、「もしAであれば、Bへ行き、またCへ行く」）を実行することができない。そして、そうした複雑な決定を実行すべくパイプライン30を設計できるが、必要と

される回路のサイズ及び複雑性はしばしばそうした設計を非現実的に為し、特にアルゴリズムが多数の種々の複雑な決定を含む場合にそうである。

#### 【0025】

結果として、プロセッサは典型的には重要な意思決定を必要とする用途において使用され、ハードウェアに組み込まれたパイプラインは殆ど意思決定が為されないか或は意思決定されない「ナンバークランチング（数値データ処理）」用途に典型的には限定される。

#### 【0026】

更には、下記に議論されるように、典型的には、特にパイプライン30が多数のPLICを含む場合、図2のパイプライン30等のハードウェアに組み込まれたパイプラインを設計/変更するよりも、図1の計算マシン10等のプロセッサに基づく計算マシンを設計/変更することが非常に易しい。 10

#### 【0027】

プロセッサ及びそれらの周辺機器（例えば、メモリ）等の計算構成要素は、典型的には、プロセッサに基づく計算マシンを形成すべくそれら構成要素の相互接続を補助する工業規格通信インターフェースを含む。

#### 【0028】

典型的には、規格通信インターフェースは2つの層、即ち、物理層及びサービス層を含む。

#### 【0029】

物理層は、回路とこの回路のインターフェース及び動作パラメータを形成する対応回路相互接続とを含む。例えば、物理層はそれら構成要素を1つのバスに接続するピンと、それらのピンから受け取ったデータをラッチするバッファと、信号をそれらピンに駆動するドライバとを含む。動作パラメータは、ピンが受け取るデータ信号の許容可能電圧範囲と、データの書き込み及び読み取りのための信号タイミングと、動作の支援されたモード（例えば、バーストモード、ページモード）とを含む。従来の物理層はトランジスタートランジスタ論理（TTL）及びRAMBUSを含む。 20

#### 【0030】

サービス層は、計算構成要素のデータ転送のためのプロトコルを含む。このプロトコルはデータのフォーマットと、構成要素によるフォーマット済みデータの送受信の方式とを含む。従来の通信プロトコルは、ファイル転送プロトコル（FTP）及びTCP/IP（拡張）を含む。 30

#### 【0031】

結果として、製造業者やその他は工業規格通信インターフェースを有する計算構成要素を典型的には設定するので、そうした構成要素のインターフェースを典型的には設計でき、それを他の計算構成要素と比較的少ない労力で相互接続することができる。これは、計算マシンの他の部分の設計に設計者自信の時間を殆ど費やすことを可能として、各種構成要素を追加或は除去することによってそのマシンを変更することを可能としている。

#### 【0032】

工業規格通信インターフェースを支援する計算構成要素を設計することは、設計ライブラリから既存の物理層を用いることによって設計時間を節約することを可能としている。これは、設計者が構成要素を既製の計算構成要素と容易にインターフェースすることを保証するものでもある。 40

#### 【0033】

そして、共通した工業規格通信インターフェースを支援する計算構成要素を用いる計算マシンを設計することは、設計者がそれら構成要素を少しの時間及び労力で相互接続することを可能としている。それら構成要素は共通インターフェースを支援するので、設計者はそれらをシステム・バスを介して少しの設計労力で相互接続することができる。そして、その支援されたインターフェースは工業規格であるので、マシンを容易に変更することができる。例えば、システム設計が進化するに伴って種々の構成要素及び周辺機器をマシンに追加することができるか、或は、テクノロジーが進化するに伴って次世代の構成要素 50

を追加／設計することが可能である。更には、構成要素が通常の工業規格サービス層を支援するので、計算マシンのソフトウェアに対応するプロトコルを具現化する既存のソフトウェア・モジュールを組み込むことができる。それ故に、インターフェース設計が本質的には既に整っているもので少しの労力で構成要素をインターフェースでき、よって、マシンに所望の機能を実行させるマシンの各種部分（例えばソフトウェア）の設計に集中することができる。

#### 【0034】

しかし残念ながら、図2のパイプライン30等のハードウェアに組み込まれたパイプラインを形成すべく、使用されるPLIC等の各種構成要素に対する既知の工業規格通信層が全くない。

10

#### 【0035】

結果として、多数のPLICを有するパイプラインを設計すべく、多大な時間を費やし、「ゼロから」種々のPLICの間の通信層を設計し且つデバッグする多大な労力を行使する。典型的には、そうしたその場限りのサービス層は種々のPLIC間で転送されるデータのパラメータに依存する。同じように、プロセッサとインターフェースするパイプラインを設計すべく、ゼロからのパイプライン及びプロセッサの間の通信層の設計及びデバッグに関して多大な時間を費やし且つ多大な労力を行使する必要がある。

#### 【0036】

同様に、そうしたパイプラインをPLICを該パイプラインに追加することによって変更すべく、典型的には、その追加されたPLICと既存のPLICとの間の通信層の設計及びデバッグに関して多大な時間を費やし且つ多大な労力を行使する。同じように、プロセッサを追加することによってパイプラインを変更すべく、或は、パイプラインを追加することによって計算マシンを変更すべく、パイプライン及びプロセッサの間の通信層の設計及びデバッグに関して多大な時間を費やし且つ多大な労力を行使しなければならないであろう。

20

#### 【0037】

結果として、図1及び図2で参照されるように、多数のPLICをインターフェースすることとプロセッサをパイプラインにインターフェースすることとの難しさのため、計算マシンを設計する際に多大な妥協を為すことがしばしば強られる。例えば、プロセッサに基づく計算マシンでは、ナンバークランピング速度を、複雑な意思決定を為す能力及び設計／変更の柔軟性と交換することを強られる。逆に、ハードウェアに組み込まれたパイプラインに基づく計算マシンでは、複雑な意思決定を為す能力と設計／変更の柔軟性を、ナンバークランピング速度と交換することを強られる。更には、多数のPLICをインターフェースすることに関する難しさのため、少数のPLICよりも多くのPLICを有するパイプラインに基づくマシンを設計することはしばしば実際的ではない。その結果、実際的なパイプラインに基づくマシンはしばしば制限された機能しか有さない。そして、プロセッサをPLICとインターフェースすることに関する難しさのため、プロセッサを1つのPLICよりも多くのPLICにインターフェースすることは実際的ではない。その結果、プロセッサ及びパイプラインを組み合わせることによって獲得される利益は最少となる。

30

40

#### 【発明の開示】

#### 【発明が解決しようとする課題】

#### 【0038】

それ故に、プロセッサに基づくマシンの意思決定を為す能力を、ハードウェアに組み込まれたパイプラインに基づくマシンのナンバークランピング速度と組み合わせることを可能とする新しい計算アーキテクチャに対する要望が生じてきている。

#### 【課題を解決するための手段】

#### 【0039】

本発明の実施例において、ピアーベクトル・マシンはホストプロセッサとハードウェアに組み込まれたパイプライン加速器とを含む。ホストプロセッサは、プログラムを実行し

50

、そしてそのプログラムに応じてホスト・データを生成し、パイプライン加速器はそのホスト・データからパイプライン・データを生成する。

【0040】

本発明の別の実施例に従えば、パイプライン加速器はパイプライン・データを生成し、ホストプロセッサはそのパイプライン・データからホスト・データを生成する。

【0041】

ビアーベクトル・マシンはプロセッサとハードウェアに組み込まれたパイプライン加速器との両方を含むので、しばしば、プロセッサだけ或はハードウェアに組み込まれたパイプラインだけを含む計算マシンよりもデータをより効率的に処理できる。例えば、加速器が数学的に集中的な演算を実行する一方で、ホストプロセッサが意思決定及び数学的に非集中的な演算を実行するようにビアーベクトル・マシンを設計できる。数学的に集中的な演算を加速器にシフトすることによって、ビアーベクトル・マシンは、しばしば、所与のクロック周波数でプロセッサだけのマシンがデータを処理する速度を凌ぐ速度でそのデータを処理できる。

【発明を実施するための最良の形態】

【0042】

図3は、本発明の一実施例に従ったビアーベクトル・アーキテクチャを有する計算マシン40の概略ブロック線図である。ホストプロセッサ42に加えて、ビアーベクトル・マシン40はパイプライン加速器44を含み、それがデータ処理の少なくとも一部を実行して、図1の計算マシン10における共同プロセッサ14の列と効果的に置き換わる。それ故に、ホストプロセッサ42及び加速器44はデータ・ベクトルを前後に転送できる「ピア」である。加速器44はプログラム命令を実行しないので、所与のクロック周波数で共同プロセッサの列ができるものよりも著しく高速にデータに対して数学的に集中的な演算を典型的には実行する。結果として、プロセッサ42の意思決定能力と加速器44のナンバークラッシュ能力とを組み合わせることによって、マシン40はマシン10等の従来の計算マシンと同一の能力を有するが、しばしばそれよりもデータをより高速に処理することができる。更には、先行して引用された「改善された計算アーキテクチャを有する計算マシン、関連システム、並びに、方法」と題された特許文献2や「改善された計算アーキテクチャ用パイプライン加速器、関連システム、並びに、方法」と題された特許文献3で議論されているように、加速器44にホストプロセッサ42と同一の通信層を設けることは、特にその通信層が工業規格である場合、マシン40の設計及び変更を補助する。そして、加速器44が多数の構成要素（例えばP L I C）を含む場合、これら構成要素にこの同一の通信層を設けることは、特にその通信層が工業規格である場合、加速器の設計及び変更を補助する。更にはマシン40も、以下に説明されると共に先行して引用された特許出願に説明されているように、他の長所等を提供し得る。

【0043】

ホストプロセッサ42及びパイプライン加速器44に加えて、ビアーベクトル計算マシン40はプロセッサ・メモリ46、インターフェース・メモリ48、バス50、ファームウェア・メモリ52、任意選択的な生データ入力ポート54、56、処理データ出力ポート58、60、並びに、任意選択的なルータ61を含む。

【0044】

ホストプロセッサ42は処理ユニット62及びメッセージ・ハンドラー64を含み、プロセッサ・メモリ46は処理ユニット・メモリ66及びハンドラー・メモリ68を含み、それらがプロセッサ・ユニット及びメッセージ・ハンドラーに対するプログラム及び作業の両メモリとしてそれぞれ役立っている。プロセッサ・メモリ46も加速器コンフィギュレーション・レジストリ70及びメッセージ・コンフィギュレーション・レジストリ72を含み、それらが、ホストプロセッサ42に加速器44の機能を構成させると共にメッセージ・ハンドラー64が生成するメッセージの構造とを構成させる各コンフィギュレーション・データを記憶する。

【0045】

パイプライン加速器 44 は少なくとも 1 つの P L I C (不図示) 上に配置されると共にハードウェアに組み込まれたパイプライン 74<sub>1</sub> ~ 74<sub>n</sub> を含み、それらがプログラム命令を実行することなしに各データを処理する。ファームウェア・メモリ 52 は加速器 44 に対するコンフィギュレーション・ファームウェアを記憶する。もし加速器 44 が多数の P L I C 上に配置されれば、それら P L I C 及びそれらの各ファームウェア・メモリは多数の回路ボード上、即ちドーターカード (不図示) 上に配置され得る。加速器 44 及びドーターカードは、先行して引用された「改善された計算アーキテクチャ用パイプライン加速器、関連システム、並びに、方法」と題された特許文献 3 や「多数パイプライン・ユニットを有するパイプライン加速器、関連計算マシン、並びに、方法」と題された特許文献 5 に更に議論されている。代替的には、加速器 44 は少なくとも 1 つの A S I C 上に配置され得て、よって構成不可能である内部相互接続を有し得る。この代替例において、マシン 40 はファームウェア・メモリ 52 を省略し得る。更には、加速器 44 が多数のパイプライン 74 を含むようにも示されているが、単一のパイプラインだけを含んでもよい。

10

【0046】

更に図 3 で参照されるように、ビアーベクトル・マシン 40 の動作は本発明の実施例に従って以下に議論される。

【0047】

＜ビアーベクトル・マシンの構成＞

ビアーベクトル・マシン 40 が先ず起動されると、処理ユニット 62 はメッセージ・ハンドラー 64 及びパイプライン加速器 44 (加速器が構成可能である場合) を構成して、マシンが所望アルゴリズムを実行するように為す。詳細には処理ユニット 62 は、以下で議論されるように、メモリ 66 に記憶されると共に、処理ユニットにメッセージ・ハンドラー 64 及び加速器 44 を構成させるホスト・アプリケーション・プログラムを実行する。

20

【0048】

メッセージ・ハンドラー 64 を構成すべく、処理ユニット 62 はレジストリ 72 からメッセージ・フォーマット情報を検索し、そのフォーマット情報をメッセージ・ハンドラーに提供して、該メッセージ・ハンドラーはこの情報をメモリ 60 に記憶する。マシン 40 が以下に議論されるようにデータを処理すると、メッセージ・ハンドラー 64 はそのフォーマット情報を用いて所望のフォーマットを有するデータ・メッセージを生成し解読する。一実施例において、フォーマット情報は拡張可能マークアップ言語 (XML) で書かれるが、他の言語或はデータ・フォーマットでも書かれ得る。処理ユニット 62 はビアーベクトル・マシン 40 が起動されるたびにメッセージ・ハンドラー 64 を構成するので、単にレジストリ 72 に記憶されたフォーマット情報を変更することによってメッセージ・フォーマットを変更することができる。代替的には、外部メッセージ・コンフィギュレーション・ライブラリ (不図示) が多数のメッセージ・フォーマットに対する情報を記憶し得て、そしてホスト・アプリケーションを設計及び/或は変更し得て、処理ユニット 62 がライブラリの選択された部分からレジストリ 72 を更新してから、更新されたレジストリからメッセージ・ハンドラー 64 に所望フォーマット情報をダウンロードするように為す。メッセージ・フォーマットとメッセージを生成して解読することとは、以下に更に議論されると共に、先行して引用された「改善された計算アーキテクチャを有する計算マシン、関連システム、並びに、方法」と題された特許文献 2 に更に議論されている。

30

40

【0049】

同様に、パイプライン加速器 44 の相互接続レイアウトを構成すべく、処理ユニット 62 はレジストリ 70 からコンフィギュレーション・ファームウェアを検索し、そのファームウェアをメッセージ・ハンドラー 64 及びバス 50 を介してメモリ 52 にダウンロードする。次いで加速器 44 は、メモリ 52 からその相互接続コンフィギュレーション・レジスタ (不図示) にファームウェアをダウンロードすることによってそれ自体を構成する。処理ユニット 62 がビアーベクトル・マシン 40 が起動されるたびに加速器 44 を構成するので、単にレジストリ 70 に記憶されたファームウェアを変更することによって、加速

50

器 4 4 の相互接続レイアウト（そして、それ故その機能と）を変更することができる。代替的には、外部加速器コンフィギュレーション・ライブラリ（不図示）は加速器 4 4 の多数のコンフィギュレーションに対するファームウェアを記憶し得て、そしてホスト・アプリケーションを設計及び／或は変更し得て、処理ユニット 6 2 がライブラリの選択された部分からレジストリ 7 0 を更新してから、更新されたレジストリからメモリ 5 2 に所望ファームウェアをダウンロードするように為す。更には、外部ライブラリ或はレジストリ 7 0 は加速器 4 4 の種々の部分及び／或は機能を規定するファームウェア・モジュールを記憶し得る。それ故、加速器 4 4 の設計及び／或は変更を補助するために、これらモジュールを使用し得る。加えて、処理ユニット 6 2 はこれらモジュールを使用し得て、マシン 4 0 がデータを処理している間に加速器 4 4 を変更する。加速器 4 4 の相互接続コンフィギュレーションとファームウェア・モジュールとは、先行して引用された「プログラマブル回路、関連計算マシン、並びに、方法」と題された特許文献 4 に更に議論されている。

【0050】

処理ユニット 6 2 も、ビアーベクトル・マシン 4 0 がデータを処理している間にパイプライン加速器 4 4 を「ソフト構成」し得る。即ち、処理ユニット 6 2 は、加速器の相互接続レイアウトを改変することなく、加速器 4 4 の機能を構成し得る。そうしたソフト・コンフィギュレーションは、以下に更に議論されると共に、先行して引用された「改善された計算アーキテクチャ用パイプライン加速器、関連システム、並びに、方法」と題された特許文献 3 に更に議論されている。

【0051】

#### ＜ビアーベクトル・マシンによるデータ処理＞

一般に、ビアーベクトル・マシン 4 0 はホストプロセッサ 4 2 及びパイプライン加速器 4 4 の間に生データの処理を効果的に分割する。例えば、ホストプロセッサ 4 2 はそのデータと関係する意思決定演算の殆ど或は全てを実行し得て、加速器 4 4 はそのデータに対する数学的に集中的な演算の殆ど或は全てを実行し得る。しかしながらマシン 4 0 は任意の所望の方式でそのデータ処理を分割する。

#### ＜ホストプロセッサの動作＞

一実施例において、ホストプロセッサ 4 2 は、ソナー・アレイ（図 5）等の遠隔装置から生データを受信し、その結果としての処理データをその遠隔装置に提供する。

【0052】

ホストプロセッサ 4 2 は、先ず、遠隔装置から入力ポート 5 4 或はバス 5 0 を介して生データを受信する。ビアーベクトル・マシン 4 0 はその受信された生データのバッファリングのための FIFO（不図示）を含み得る。

【0053】

次に、処理ユニット 6 2 はパイプライン加速器 4 4 による処理のための生データを準備する。例えば、ユニット 6 2 は、例えば、生データの何れを加速器 4 4 に送信すべきか、或は、何れのシーケンスでその生データを送信すべきかを決定し得る。或は、ユニット 6 2 は生データを処理し得て、加速器 4 4 に送信するための中間データを生成する。生データの準備は、先行して引用された「改善された計算アーキテクチャを有する計算マシン、関連システム、並びに、方法」と題された特許文献 2 に更に議論されている。

【0054】

生データを準備している間、処理ユニット 5 4 は 1 つ或はそれ以上の「ソフト・コンフィギュレーション」コマンドを生成して、加速器 4 4 の機能を変更し得る。マシン 4 0 が起動された際に加速器 4 4 の相互接続レイアウトを構成するファームウェアとは異なり、ソフト・コンフィギュレーション・コマンドはその相互接続レイアウトを改変することなしに加速器の機能を制御する。例えば、ソフト・コンフィギュレーション・コマンドは、加速器 4 4 が処理するデータ・ストリングのサイズ（例えば、32 ビット或は 64 ビット）を制御し得る。加速器 4 4 のソフト・コンフィギュレーションは、先行して引用された「改善された計算アーキテクチャ用パイプライン加速器、関連システム、並びに、方法」と題された特許文献 3 に更に議論されている。

## 【0055】

次いで処理ユニット62はその準備されたデータ及び／或はソフト・コンフィギュレーション・コマンド（単数或は複数）をインターフェース・メモリ48の対応する箇所にロードし、それがユニット62及び加速器44の間のFIFOバッファとして作用する。

## 【0056】

次に、メッセージ・ハンドラー64はインターフェース・メモリ48から準備されたデータ及び／或はソフトウェア・コマンド（単数或は複数）を検索して、データ及び／或はコマンド（単数或は複数）、並びに、関連情報を含むメッセージ・オブジェクトを生成する。典型的には加速器44は、データ／コマンド（単数或は複数）及び関連情報（集合的には「情報」）、即ち、a)情報の意図された仕向先（例えば、パイプライン74<sub>1</sub>）、  
b)優先順位（例えば、加速器は先行して受信されるデータの前或は後にそのデータを処理すべきか）、c)メッセージ・オブジェクトの長さ或は終端、並びに、d)データの固有例（例えば、一千のセンサから成るアレイからのセンサ信号番号9）を記述する4つの識別子が必要である。この決定を補助すべく、メッセージ・ハンドラー64は、先に議論されたように、所定のフォーマットを有するメッセージ・オブジェクトを生成する。準備されたデータ／ソフト・コンフィギュレーション・コマンド（単数或は複数）に加えて、メッセージ・オブジェクトは、典型的には、4つの先に記載した識別子を含むと共にオブジェクトが含む情報のタイプを記述する識別子（例えば、データ・コマンド）をも含み得るヘッダーと、データが処理されることになるアルゴリズムとを含む。この後者の識別子は、仕向先パイプライン74が多数のアルゴリズムを具現化する場合に有用である。ハンドラー64はインターフェース・メモリ48からヘッダー情報を検索し得るか、準備されたデータ或はコマンド（単数或は複数）を検索するインターフェース・メモリ内の箇所に基づいてヘッダーを生成し得る。メッセージ・ヘッダーを解説することによって、ルータ61及び／或は加速器44はメッセージ・オブジェクト内の情報を所望の仕向先に送ることができ、その仕向先にその情報を所望の順序で処理させ得る。

## 【0057】

メッセージ・オブジェクトを生成するための代替実施例が存在する。例えば、各メッセージ・オブジェクトがデータ或はソフト・コンフィギュレーション・コマンドの何れかを含むと説明されているが、単一のメッセージ・オブジェクトはデータと1つ或はそれ以上のコマンドとの両方を含み得る。更には、メッセージ・ハンドラー64がインターフェース・メモリ48からデータ及びコマンドを受信すると説明されているが、処理ユニット54からデータ及びコマンドを直接的に受信し得る。

## 【0058】

メッセージ・オブジェクトの生成は、先行して引用された「改善された計算アーキテクチャを有する計算マシン、関連システム、並びに、方法」と題された特許文献2に更に議論されている。

## ＜パイプライン加速器＞

パイプライン加速器44はメッセージ・ハンドラー64からメッセージ・オブジェクトを受信し解説して、そのオブジェクト内のデータ及び／或はコマンドを所望の仕向先に効果的に送る。この技術は、処理ユニット62及びパイプライン74によって具現化されたアルゴリズムの数が比較的小さい場合に特に有用であり、よってルータ61は省略され得る。代替的には、処理ユニット62或は番号パイプライン74によって具現化されたアルゴリズムの数が比較的大きい場合、ルータ61はメッセージ・ハンドラー64からメッセージ・オブジェクトを受信し解説して、そのオブジェクト内のデータ及び／或はコマンドを加速器44内の所望の仕向先に効果的に送る。

## 【0059】

少数の処理ユニット・アルゴリズム及びパイプライン74が存在する場合の一実施例において、各パイプラインは、同時に、メッセージ・オブジェクトを受信し、ヘッダーを分析して、それが意図されたメッセージの受取人であるか否かを決定する。もしメッセージ・オブジェクトが特定のパイプライン74に意図されていれば、そのパイプラインはその

メッセージを解説し、回復されたデータ／コマンド（単数或は複数）を処理する。しかしながら、もしメッセージ・オブジェクトが特定のパイプライン 7 4 に意図されていなければ、そのパイプラインはそのメッセージ・オブジェクトを無視する。例えば、メッセージ・オブジェクトがパイプライン 7 4<sub>1</sub>による処理のためのデータを含むと仮定する。それ故に、パイプライン 7 4<sub>1</sub>はメッセージ・ヘッダーを分析し、それがそのデータの意図された仕向先であることを決定し、メッセージからデータを回復し、その回復されたデータを処理する。逆に、パイプライン 7 4<sub>2</sub>〜7 4<sub>n</sub>の各々がメッセージ・ヘッダーを分析し、それがそのデータの意図された仕向先ではないことを決定し、よってそのデータの回復或は処理を為さない。もしメッセージ・オブジェクト内のデータが多数のパイプライン 7 4 に意図されていれば、メッセージ・ハンドラー 6 4 は同一データを含む各メッセージ・オブジェクトから成るシーケンスを、各仕向先パイプラインに対して 1 つのメッセージで、生成して送信する。代替的には、メッセージ・ハンドラー 6 4 は、仕向先パイプラインの全てを識別するヘッダーを有する単一のメッセージ・オブジェクトを送信することによって、データを仕向先パイプライン 7 4 の全てに同時に送信し得る。メッセージ・オブジェクトからデータ及びソフト・コンフィギュレーション・コマンドを回復することは、先行して引用された「改善された計算アーキテクチャ用パイプライン加速器、関連システム、並びに、方法」と題された特許文献 3 に更に議論されている。

#### 【0060】

多数の処理ユニット処理或はパイプライン 7 4 が存在する場合の別の実施例において、各パイプラインはルータ 6 1 からメッセージ・オブジェクトを受信する。ルータ 6 1 がメッセージ・オブジェクトを目標パイプライン 7 4 だけに理想的には送信すべきであるが、その目標パイプラインもヘッダーを分析して、それが意図されたメッセージの受取人であるか否かを決定する。そうした分析は潜在的なメッセージ・ルーチン・エラー、即ち例外を識別する。もしメッセージ・オブジェクトが目標パイプライン 7 4 に意図されていれば、そのパイプラインはそのメッセージを解説し、回復されたデータ／コマンド（単数或は複数）を処理する。しかしながら、もしメッセージ・オブジェクトがその目標パイプライン 7 4 に意図されていなければ、そのパイプラインはそのメッセージ・オブジェクトに対しての処理を無視して、更にルーチン例外が発生したことを示す新メッセージをホストプロセッサ 4 2 に発し得る。ルーチン例外の取り扱いは、先行して引用された「改善された計算アーキテクチャを有する計算マシン、関連システム、並びに、方法」と題された特許文献 2 に議論されている。

#### 【0061】

次に、パイプライン加速器 4 4 はメッセージ・オブジェクトから回復された入来データ及び／或はコマンドを処理する。

#### 【0062】

データに対して、仕向先パイプライン或は複数の仕向先パイプライン 7 4 はデータに対する各演算或は複数演算を実行する。図 2 と連携して議論されたように、パイプライン 7 4 はプログラム命令を実行しないので、しばしばデータをパイプライン・クロックの周波数と略同一である速度で処理できる。

#### 【0063】

第 1 実施例において、単一パイプライン 7 4 は入来データを処理することによって結果としてのデータを生成する。

#### 【0064】

第 2 実施例において、多数のパイプライン 7 4 は入来データを順次処理することによって結果としてのデータを生成する。例えば、パイプライン 7 4 は入来データに対して第 1 演算を実行することによって第 1 中間データを生成し得る。次に、パイプライン 7 4<sub>2</sub>はその第 1 中間データに対して第 2 演算を実行することによって第 2 中間データを生成し得て、等々であり、そのチェーン中の最終パイプライン 7 4 が結果データを生成するまで同様である。

#### 【0065】



第3実施例において、多数のパイプライン74は入力データを並列して処理することによって結果としてのデータを生成する。例えば、パイプライン74<sub>1</sub>は入力データから成る第1組に対して第1演算を実行することによって結果としてのデータから成る第1組を生成し得る。同時に、パイプライン74<sub>2</sub>は入力データから成る第2組に対して第2演算を実行することによって結果としてのデータから成る第2組を生成し得る、等々である。

【0066】

代替的には、先の3つの実施例の任意の組み合わせに従って、パイプライン74は入力データから結果としてのデータを生成し得る。例えば、パイプライン74<sub>1</sub>は入力データから成る第1組に対して第1演算を実行することによって結果としてのデータから成る第1組を生成し得る。同時に、パイプライン74<sub>2</sub>及び74<sub>3</sub>は入力データから成る第2組に対して第2及び第3の演算を順次実行することによって結果としてのデータから成る第2組を生成し得る。

【0067】

先の実施例及び代替実施例の内の任意のものにおいて、単一のパイプライン74は多数の演算を実行し得る。例えば、パイプライン74<sub>1</sub>はデータを受信し得て、その受信データに対して第1演算を実行することによって第1中間データを生成し得て、その第1中間データを一時的に記憶し得て、その第1中間データに対して第2演算を実行することによって第2中間データを生成し得て、等々であり、結果データを生成するまで同様である。パイプライン74<sub>1</sub>に第1演算実行から第2演算実行までスイッチさせる等々の多数の技術がある。そうした技術は、先行して引用された「改善された計算アーキテクチャ用パイプライン加速器、関連システム、並びに、方法」(代理人整理番号1934-13-3)と題された特許文献3に議論されている。

【0068】

ソフト・コンフィギュレーション・コマンドに対して、加速器44はメッセージ・ヘッダーによって示される対応するソフト・コンフィギュレーション・レジスタ(半数或は複数)(不図示)におけるビットを設定する。先に議論されたように、これらビットを設定することは、典型的には、加速器44の機能を変えるものであり、その相互接続レイアウトを変えることがない。これは、例えば、入力ピン或は出力ピンとしての外部ピンを設定する、或は、アドレス指定モードを選択することに対するプロセッサの制御レジスタにおけるビットを設定することと同様である。更には、ソフト・コンフィギュレーション・コマンドはデータを保持するためにレジスタ或はテーブル(レジスタのアレイ)を仕切ることができる。加速器44によって実行される別のソフト・コンフィギュレーション・コマンド或は演算はデータをソフト構成されたレジスタ或はテーブルにロードし得る。加速器44のソフト・コンフィギュレーションは、先行して引用された「改善された計算アーキテクチャ用パイプライン加速器、関連システム、並びに、方法」(代理人整理番号1934-13-3)と題された特許文献3に更に議論されている。

【0069】

次に、パイプライン加速器44は結果としてのデータを、更なる処理のためにルータ61を介して(或はもしそのルータが省略されていれば直接的に)、ホストプロセッサ42に提供する。

【0070】

代替的には、パイプライン加速器44は結果としてのデータを、出力ポート60を介して直接的に、或は、ルータ61(もしあれば)、バス50、ホストプロセッサ42、並びに、出力ポート58を介して間接的に、の何れかで、遠隔仕向先(図5)に提供する。結果として、この代替実施例において、加速器44によって生成された結果としてのデータは最終処理データである。

【0071】

加速器44が結果としてのデータを(更なる処理のため或は遠隔装置まで通過させるため(図5))ホストプロセッサ42に提供するときには、それはメッセージ・ハンドラー64によって生成されたメッセージ・オブジェクトと同一のフォーマットを有するメッセ

ージ・オブジェクトにおいてこのデータを送信する。メッセージ・ハンドラー64によって生成されたメッセージ・オブジェクトと同じように、加速器44によって生成されたメッセージ・オブジェクトは、例えば、結果としてのデータの仕向先及び優先順位を特定するヘッダーを含む。例えば、そのヘッダーはメッセージ・ハンドラー64に、その結果としてのデータを遠隔装置までポート58を介して通過させるように命令し得るか、或は、データの処理を制御することになるのは処理ユニット62によって実行されるプログラムの何れの部分かを特定し得る。同一メッセージ・フォーマットを用いることによって、加速器44はホストプロセッサ42と同一のインターフェース層を有する。これは、特にインターフェース層が工業規格であればピア・ベクトル・マシン40の設計及び変更を補助する。

10

#### 【0072】

パイプライン加速器44及びパイプライン66の構造及び動作は、先行して引用された「改善された計算アーキテクチャ用パイプライン加速器、関連システム、並びに、方法」（代理人整理番号1934-13-3）と題された特許文献3に更に議論されている。

#### 【0073】

＜ホストプロセッサによるパイプライン加速器からの受信及び処理＞

加速器44からメッセージ・オブジェクトを受信すると、メッセージ・ハンドラー64は先ずそのメッセージ・ヘッダーを解説して、回復されたデータを指定された仕向先に送る。

#### 【0074】

もしそのヘッダーがデータがポート58を介して遠隔装置（図5）に渡されることを示せば、メッセージ・ハンドラー64はそのデータをポート58に直接的に、又は、インターフェース・メモリ48或は、別のメモリ内に形成されたポートFIFOバッファ（不図示）に、提供してから、そのバッファからポート58に、向かわせる。多数のポート58と多数の遠隔装置も意図されている。

20

#### 【0075】

しかしながら、もしヘッダーが処理ユニット62がデータを更に処理することになっていることを示せば、メッセージ・ハンドラー62はそのデータの処理を制御することになる処理ユニット・プログラムの部分と対応するインターフェース・メモリ48の箇所に該データを記憶する。より詳細には、同一ヘッダーが処理ユニット54によって実行されるプログラムの何れの部分（単数或は複数）がそのデータの処理を制御することになるのかを間接的に示す。結果として、メッセージ・ハンドラー64はそのプログラム部分と対応しているインターフェース・メモリ48の箇所（FIFO等）にそのデータを記憶する。

30

#### 【0076】

先に議論されたように、インターフェース・メモリ48は加速器44及び処理ユニット62の間のバッファとして作用し、よって処理ユニットが加速器と同期されない際にデータの転送を可能としている。例えば、同期に関するこの欠如は、加速器44が処理ユニット62よりもデータを高速に処理する際に生じ得る。インターフェース・メモリ48を用いることによって、加速器44は処理ユニット62のより遅い応答によって遅くさせられない。これは、割り込みの取り扱いに対する確定できない応答時間と関連する非効率というペナルティをも回避する。加速器44の出力メッセージの処理ユニット62による確定できない取り扱いは、a) バックアップされた出力メッセージに対する記憶及び取り扱い、或は、b) バックアップされたメッセージが上書きされることを防止するためのパイプライン全体にわたるアイドルリング制御、の何れかを提供させるように、設計者に強制することによって加速器の設計を不必要に複雑化することになる。それ故に、加速器44及び処理ユニット62の間のバッファとして作用するインターフェース・メモリ48の使用は、a) 加速器が設計に対してより容易、b) 加速器がより少ない下部組織を必要として、大きなPLICアプリケーションを保持できる、c) 加速器が合理化され得て、出力データがより緩慢なプロセッサによって「ブロック」されないのにより高速に作動する等の幾つかの所望の結果を有する。

40

50

## 【0077】

次いで、メッセージ・ハンドラー64がインターフェース・メモリ48に記憶したデータに対して、処理ユニット62はそのインターフェース・メモリからそのデータを検索する。処理ユニット62はインターフェース・メモリ48にポーリングし得て、新データが特定箇所にいつ到達したかを決定するか、或は、メッセージ・ハンドラー64は処理ユニットにデータの到達を通知する割り込み若しくは他の信号を生成し得る。一実施例において、処理ユニット62がデータを検索する前、メッセージ・ハンドラー64はそのデータを含むメッセージ・オブジェクトを生成する。より詳細には、メッセージ・オブジェクト内のデータを受信するために、処理ユニット62によって実行されるプログラムを設計し得る。それ故にメッセージ・ハンドラー64は、データだけを記憶する代わりに、インターフェース・メモリ48内にメッセージ・オブジェクトを記憶し得る。しかし、メッセージ・オブジェクトは、典型的には、それが含有するデータよりも著しく大きなメモリ・スペースを占有する。結果として、メモリを節約すべく、メッセージ・ハンドラー64はパイプライン加速器44からのメッセージ・オブジェクトを解説し、そのデータをメモリ48に記憶してから、処理ユニット62がそのデータを受信する準備が為されるとメッセージ・オブジェクトを効果的に再生成する。次いで、処理ユニット62はそのメッセージ・オブジェクトを解説し、メッセージ・ヘッダーにおいて識別されたプログラム部分の制御の下、そのデータを処理する。

10

## 【0078】

次に、プロセッサ・ユニット62はプログラムの仕向先部分の制御下で検索データを処理し、処理データを生成し、その処理データを該処理データの意図された仕向先と対応するインターフェース・メモリ48の箇所に記憶する。

20

## 【0079】

次いで、メッセージ・ハンドラー64は処理データを検索し、それを指定された仕向先に提供する。処理データを検索すべく、メッセージ・ハンドラー64はメモリ48にポーリングし得て、データがいつ到達したかを決定するか、或は、処理ユニット62はメッセージ・ハンドラーに割り込み或は他の信号でデータの到達を通知し得る。処理データをその意図された仕向先に提供すべく、メッセージ・ハンドラー64はそのデータを含むメッセージ・オブジェクトを生成し得て、そのメッセージ・オブジェクトをデータの更なる処理のために加速器44に送り戻す。或は、ハンドラー64はデータをポート58に送信し得るか、或は、処理ユニット62による更なる処理のためにメモリ48の別の箇所に送信し得る。

30

## 【0080】

ホストプロセッサのパイプライン加速器44からのデータの受信及び処理は、先行して引用された「改善された計算アーキテクチャを有する計算マシン、関連システム、並びに、方法」(代理人整理番号1934-12-3)と題された特許文献2に更に議論されている。

## 【0081】

＜ビアーベクトル・マシンを用いての代替的データ処理技術＞

更に図3で参照されるように、ホストプロセッサ44がデータを受信し処理してから、そのデータを更なる処理のためにパイプライン加速器44に送信する先に記載された実施例に対する代替例が存在する。

40

## 【0082】

代替例において、ホストプロセッサ44はデータの少なくとも幾つかに対する処理の余てを実行し、よってこのデータは更なる処理のためにパイプライン加速器44に送信されない。

## 【0083】

別の代替例において、パイプライン加速器44はポート56を介して遠隔装置(図5)から生データを直接的に受信し、その生データを処理する。加速器44は、次いで、その処理データをポート60を介して遠隔装置に直接的に送信し戻し得るか、或は、処理デー

50

タを更なる処理のためにホストプロセッサ 42 に送信し得る。後者の場合、加速器 44 は先に議論されたようにそのデータをメッセージ・オブジェクト内にカプセル化し得る。

【0084】

更に別の代替例において、加速器 44 は、ハードウェアに組み込まれたパイプライン 74 に加えて、デジタル信号プロセッサ (DSP) 等の 1 つ或はそれ以上の命令実行プロセッサを含み得て、そのパイプラインのナンバークランピング能力を補足する。

【0085】

#### ＜ビームベクトル・マシンの具現化例＞

更に図 3 で参照されるように、一実施例において、パイプライン・バス 50 は規格 133 MHz PCI バスであり、パイプライン 74 は 1 つ或はそれ以上の規格 PMC カード上に含まれ、メモリ 52 は各々が各 PMC カード上に位置決めされている 1 つ或はそれ以上のフラッシュメモリである。

【0086】

#### ＜ビームベクトル・マシンの適用例＞

図 4 は、本発明の実施例に従った、図 3 のビームベクトル・マシン 40 を組み入れるソナー・システム 80 のブロック線図である。マシン 40 に加えて、システム 80 は、ソナー信号を受信し伝送するための変換要素 84、—84、から成るアレイ 82、デジタル—アナログ変換器 (DAC) 86、—86、アナログ—デジタル変換器 (ADC) 88、—88、並びに、データ・インターフェース 90 を含む。ソナー信号を生成し処理することがしばしば数学的に集中的な機能であるので、マシン 40 はしばしばこれらの機能を、図 3 と連携されて先に議論されたように、従来の計算マシン (多数プロセッサ・マシン 10 (図 1) 等) が所与のクロック周波数でできるものよりも、より迅速に且つ効率的に実行できる。

【0087】

動作の送信モード中、アレイ 82 はソナー信号を水 (不図示) 等の媒体に送信する。先ず、ビームベクトル・マシン 40 はポート 92 で受信した生信号データを、アレイ要素 84 の各々に 1 つずつとなるように、 $n$  個のデジタル信号に変換する。これら信号の大きさ及び位相はアレイ 82 の伝送ビーム・パターンを指図する。次に、マシン 40 はこれらデジタル信号をインターフェース 90 に提供し、該インターフェースがそれら信号を各アナログ信号への変換のために各 DAC 86 に提供する。例えばインターフェース 90 はマシン 40 からデジタル信号を順次受信するバッファとして作用し得て、それら信号の全  $n$  個を受信しバッファするまでそれらを記憶してから、それら順次的な信号サンプルを各 DAC 86 に同時に提供する。次いで、変換要素 84 はそれらアナログ信号を各音波に変換して、相互に干渉させて、ソナー信号のビームを形成する。

【0088】

動作の受信モード中、アレイ 82 は媒体 (不図示) からソナー信号を受信する。受信されたソナー信号は遠隔オブジェクトによって反射された伝送ソナー信号の部分と、環境及び遠隔オブジェクトによって放出された音響エネルギーとで構成されている。先ず、変換要素 84 はソナー信号を構成する各音波を受信し、それら音波を  $n$  個のアナログ信号に変換し、それらアナログ信号を ADC 88 に提供して  $n$  個の各デジタル信号に変換する。次に、インターフェース 90 はこれらデジタル信号をビームベクトル・マシン 40 に処理のために提供する。例えば、インターフェース 90 は ADC 88 からデジタル信号を並列に受信するバッファとして作用し得て、それら信号をマシン 40 に順次提供する。マシン 40 がデジタル信号に実行する処理はアレイ 82 の受信ビーム・パターンを指図する。フィルタリング、帯域シフト、スペクトル変換 (例えば、フーリエ変換)、並びに、回転等の付加的な処理ステップはそれらデジタル信号に適用される。次いでマシン 40 は処理された信号データを、ポート 94 を介して、位置決めされたオブジェクトを視認するための表示装置等の別の装置に提供する。

【0089】

ソナー・システム 80 と連携して議論されたが、ソナー・システム以外の各種システム

もビアーベクトル・マシン 40 を組み入れることができる。

【0090】

先行する議論は当業者が本発明を作製し使用することを可能とすべく提示されている。種々実施例への様々な変更は当業者には容易に明かであろうし、ここでの包括的な原則は本発明の精神及び範囲から逸脱することなしに他の実施例及び適用例に適用され得る。よって、本発明は図示された実施例に限定されることが意図されておらず、ここに開示された原理及び特徴と一貫した最も広い範囲と一致されるべきものである。

【図面の簡単な説明】

【0091】

【図1】図1は、従来の多数プロセッサ・アーキテクチャを有する計算マシンのブロック線図である。 10

【図2】図2は、従来のハードウェアに組み込まれたパイプラインのブロック線図である。

【図3】図3は、本発明の実施例に従ったビアーベクトル・アーキテクチャを有する計算マシンのブロック線図である。

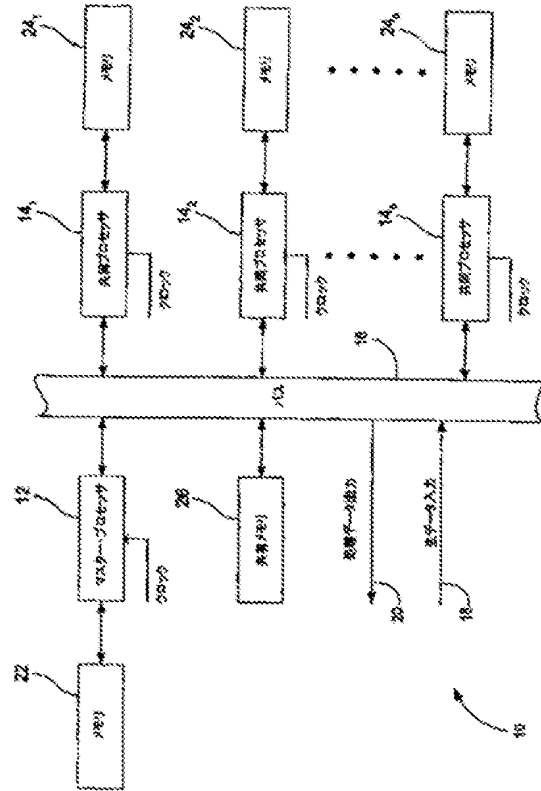
【図4】図4は、本発明の実施例に従った図3のビアーベクトル計算マシンを組み入れている電子システムの概略ブロック線図である。

【符号の説明】

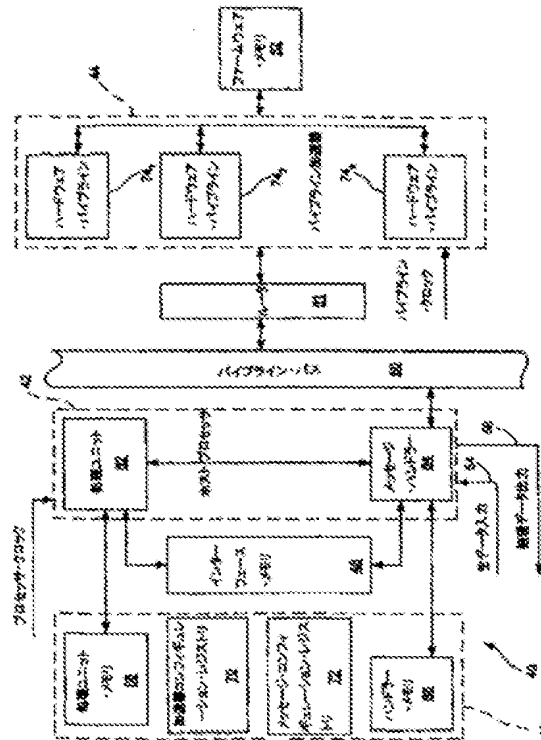
【0092】

10	計算マシン	20
14	共同プロセッサ	
40	ビアーベクトル・マシン	
42	ホストプロセッサ	
44	パイプライン加速器	
46	プロセッサ・メモリ	
48	インターフェース・メモリ	
50	パイプライン・バス	
52	ファームウェア・メモリ	
54	生データ入力ポート	
58	処理データ出力ポート	30
61	ルータ	
62	処理ユニット	
64	メッセージ・ハンドラー	
66	処理ユニット・メモリ	
68	ハンドラー・メモリ	
70	加速器コンフィギュレーション・レジストリ	
72	メッセージ・コンフィギュレーション・レジストリ	
74	ハードウェアに組み込まれたパイプライン	
80	ソナー・システム	
84	要素	40
86	デジタル-アナログ変換器	
88	アナログ-デジタル変換器	
90	データ・インターフェース	

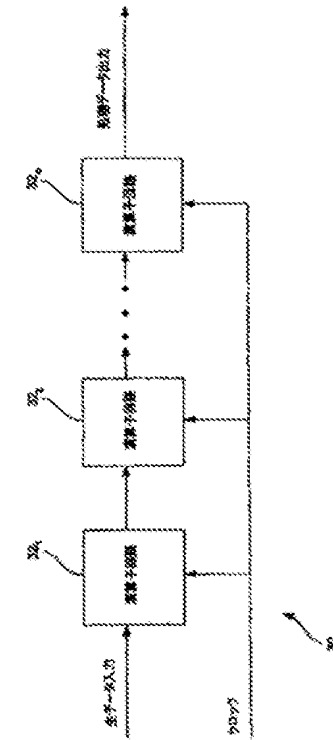
【図 1】



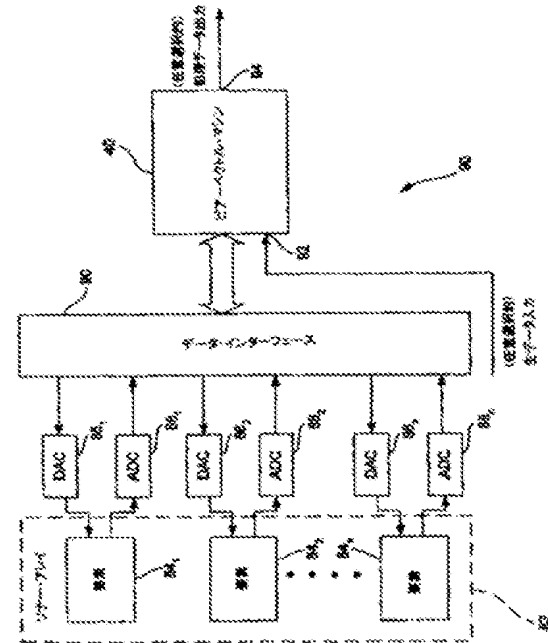
【図 3】



【図 2】



【図 4】



【国際調査報告】

INTERNATIONAL SEARCH REPORT		Inventor Application No. PCT/US 03/34557
<b>A. CLASSIFICATION OF SUBJECT MATTER</b> IPC 7 606F9/38 606F15/78		
According to International Patent Classification (IPC) or to both national classification and IPC		
<b>B. FIELDS SEARCHED</b> Minimum documentation searched (classification system followed by classification symbols) IPC 7 606F		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practical, search terms used) EPO-Internal, WPI Data, PAJ, INSPEC		
<b>C. DOCUMENTS CONSIDERED TO BE RELEVANT</b>		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 945 708 A (TEXAS INSTRUMENTS INC) 29 September 1999 (1999-09-29) page 1	1-27
X	EP 1 061 438 A (HEWLETT PACKARD CO) 20 December 2000 (2000-12-20) page 4	1-27
X	LECURIEUX-LAFAYETTE G: "UN SEUL FPGA DOPE LETRAITEMENT D'IMAGES" ELECTRONIQUE, CEP COMMUNICATION, PARIS, FR, no. 55, 1996, pages 98,101-103, XP000551946 ISSN: 1157-1152 the whole document	1-27
-/-		
<input checked="" type="checkbox"/> Further documents are listed in the continuation of box C. <input checked="" type="checkbox"/> Patent family members are listed in annex.		
* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier document (as published on or after the international filing date) "L" document which may throw doubts on priority claims or which is cited to establish the publication date of another claim or other special reason (as specified) "C" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "Z" document member of the same patent family		
Date of the actual completion of the international search 10 December 2004		Date of mailing of the international search report 21/12/2004
Name and mailing address of the ISA European Patent Office, P.O. 8016 Patenthaus 2 NL - 2200 HW Rijswijk Tel. (+31-70) 340-2040, Tx. 31 861 epo nl, Fax (+31-70) 340-2016		Authorized officer Karls, S

Form PCT/ISA/210 (second sheet) (January 2004)

## INTERNATIONAL SEARCH REPORT

International Application No.  
PCT/US 03/34557

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5 583 964 A (WANG SHAY-PING T) 10 December 1996 (1996-12-10) abstract	1-4, 16, 18-21
X	VERMEULEN F ET AL: "Flexible hardware acceleration for multimedia oriented microprocessors" MICRO-33. PROCEEDINGS OF THE 33RD. ANNUAL ACM/IEEE INTERNATIONAL SYMPOSIUM ON MICROARCHITECTURE. MONTEREY, CA, DEC. 10 - 13, 2000, PROCEEDINGS OF THE ANNUAL ACM/IEEE INTERNATIONAL SYMPOSIUM ON MICROARCHITECTURE, LOS ALAMITOS, CA : IEEE COMP. SOC, US. 10 December 2000 (2000-12-10), pages 171-177, XP010528885 ISBN: 0-7695-0924-X page 1	1, 16, 18

Form PCT/ISA/C19 (continuation of second sheet) (January 2005)



## INTERNATIONAL SEARCH REPORT

Information on patent family members

 No. International Application No.  
 PCT/US 03/34557

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0945788	A	29-09-1999	DE 69919059 D1
			EP 0945788 A2
			JP 11272631 A
			US 6256724 B1
EP 1061438	A	20-12-2000	EP 1061438 A1
			EP 1104561 A1
			WO 0077626 A1
			JP 2003501775 T
US 5583964	A	10-12-1996	AU 2126995 A
			CA 2189148 A1
			CN 1150847 A
			DE 19581638 C2
			DE 19581638 T0
			DE 29521338 U1
			GB 2302192 A
			WO 9530194 A1
			US 5740325 A

Form PCT/ISA/210 (patent family areas) (January 2004)

## フロントページの続き

(31)優先権主張番号 10/684,053

(32)優先日 平成15年10月9日(2003.10.9)

(33)優先権主張国 米国(US)

(31)優先権主張番号 10/684,057

(32)優先日 平成15年10月9日(2003.10.9)

(33)優先権主張国 米国(US)

(31)優先権主張番号 10/684,102

(32)優先日 平成15年10月9日(2003.10.9)

(33)優先権主張国 米国(US)

(81)指定国 AP(BW, GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), EA(AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), EP(AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), OA(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG), AE, AC, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, GR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW

(74)代理人 100135585

弁理士 西尾 務

(72)発明者 マートウル, チャンダン

アメリカ合衆国 バージニア州 20109 マナサッス, プライベートィス コート 11162

(72)発明者 ヘレンバツハ, スコット

アメリカ合衆国 バージニア州 20106 アメッスビル, クアイル リッジ ドライブ 15381

(72)発明者 ラーブ, ジョン, ダブリュー,

アメリカ合衆国 バージニア州 20110 マナサッス, リバー クレスト ロード 9350

(72)発明者 ジャクソン, ラリー

アメリカ合衆国 バージニア州 20112 マナサッス, クレストブロッック ドライブ 13093

(72)発明者 ジョウンス, マーク

アメリカ合衆国 バージニア州 20120 セントレビル, オークマー プレイス 15342

(72)発明者 カーサロ, トロイ

アメリカ合衆国 バージニア州 22701 カルベバー, ケストラル コート 1524